netlink and go

simple webserver

```go
func handler(w http.ResponseWriter, r
*http.Request) {
    fmt.Fprintf(w, "Refreshing Memories")
}



func main() {
    http.HandleFunc("/", handler)
    log.Fatal(http.ListenAndServe(":8080",
nil))
}
```

simple webserver

```go
func handler(w http.ResponseWriter, r
*http.Request) {
    fmt.Fprintf(w, "Refreshing Memories")
}



func main() {
    http.HandleFunc("/", handler)
    log.Fatal(http.ListenAndServe(":8080",
nil))
}
```
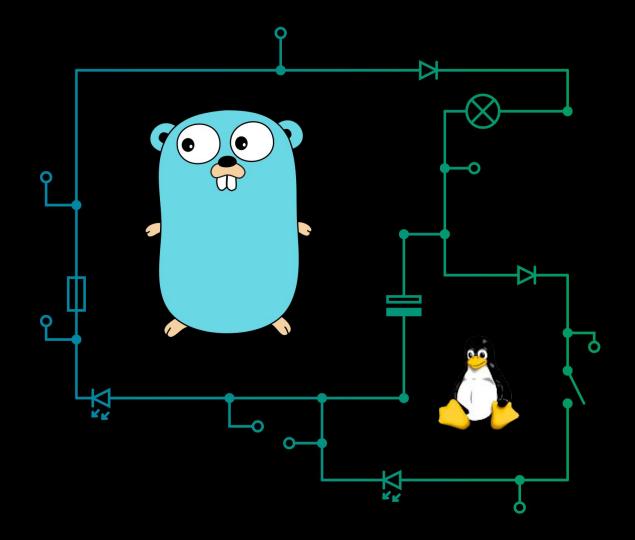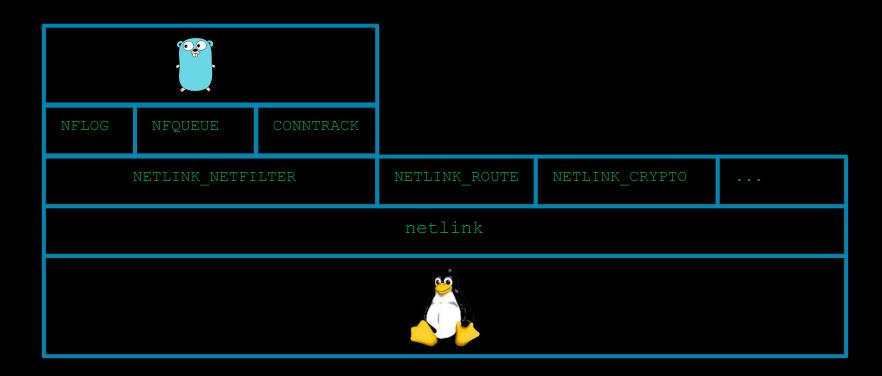
Monitoring?

netlink family

NFLOG	NFQUEUE	CONNTRACK
NETLINK_NETFILTER		

| NETLINK_ROUTE | NETLINK_CRYPTO | ... |

netlink

the binary protocol

```
    <------- NLA_HDRLEN ------> <-- NLA_ALIGN(payload)-->
   +--------------------+- - -+- - - - - - - - - - -+- - -+
   |       Header       | Pad |      Payload        | Pad |
   |   (struct nlattr)  | ing |                     | ing |
   +--------------------+- - -+- - - - - - - - - - -+- - -+
```

netlink_nflog example

```go
func main() {
    config := nflog.Config{
        Group: 100,
    }

    nf, _ := nflog.Open(&config)

    ctx, _ := context.WithTimeout(context.Background(), 30*time.Second)

    fn := func(m nflog.Msg) int {
        fmt.Printf("%v\n", m)
        Return 0
    }

    nf.Register(ctx, fn)

    <-ctx.Done()
}
```

thank you

sources

```
$ man 7 netlink
```

https://github.com/florianl/go-nflog

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/include/uapi/linux/netlink.h